operates at 2 kbps (5 octets per 20 ms).

In Figure 5 a mapping table 12 is shown which is to be used together with a fixed size length field 11 in accordance with the invention. As appears from the table the code values do not correspond to the mini cells sizes but instead predefined cell sizes are allocated to a respective code value only three code bits are used. Examples of mini cells sizes are given in the size column of the mapping list. The sizes vary from 4 to 60 octets. Of course the range can be increased, but the maximum number of sizes is given by the number of code bits used.

To expand the number of sizes that can be used together with the non-linear coding it is possible to extend the fixed LEN field 11 on demand. Two methods will be described. Either an extension bit in the fixed size LEN field 11 is used as a qualifier for extension of the LEN field 11 and the method is referred to as the extension bit method, or is one of the length field codes used as qualifier for extension of the LEN field 11 in which case the method is referred to as the extension code method.

In Figure 6 a bit 13, also labeled E, following the LEN field 11 is reserved as an extension bit 13. When the extension bit 13 is set to 1 this will indicate that the header of the mini cell comprises an extended LEN field 14 of the same size as the fixed size LEN field. When the extension bit is zero, the cell header comprises the fixed LEN field 11 only.

The extended length field 14 comprises 3 bits in the illustrated example.

When the extension bit 13 is set the number of bits available for the mapping table 12 will increase from 3 to 6 bits leaving a mapping table 15 shown in Figure 7. Since the extension bit 13 is reserved for this purpose it cannot be used for code size mapping purposes.

A variation of the extension bit method is to append an extension bit 11B to the extended LEN field 14. The appended extension field is used to indicate if there is a further extended LEN field in the header in the mini cell or not. If the appended extension bit 11B is set to 1 this indicates that a second extended length field 14A should be added to the header, thus increasing the number of code bits in table 15 from 6 to 9. If the appended extension field comprises a bit which is set to 0 no such second field is used.

In Figure 8 the extension code method is illustrated. In accordance with this method a code in the fixed length field 11 of Figure 4 is reserved and is used as extension code. Suppose, as an example, that binary code 111 in mapping table 12 is used as an extension code. When this code 111 is present in the fixed length field 11 it means that an extended length field 14 should be included in the header of the mini cell. Thus another 3 bits are available for size mapping. This has been illustrated in Figure 8. This method will reduce the number of sizes in mapping table 12 with 1 and will add another seven cell sizes that can be mapped on the additional 8 code values of the extended length field 14.

From band width efficiency view the extension code method is better than the extension bit method since it requires 3 bits, while the extension bit method requires 4 bits. Looking on the value range the extension bit method is better than the extension code method since it provides 16 different cell sizes compared to 14 as provided by the extension code method.

In Figure 9 the extension bit method has been combined with the extension code method in a manner that allows for high efficiency use of the bits available in a cell header while at the same time a broad range of cell sizes are covered and the band width is used efficiently.

The basic format of the mini cell using this combined coding method is shown in Figure 9. The mini cell comprises a header 21 of 2 octets and a payload part 22 which may comprise

from 1 to 48 octets. The four least significant bits of the length of the mini cell is indicated in a small fixed size length field 23, LEN field, in the header. The LEN field 23 comprises 4 bits. The header also comprises a CID field 24 which occupies 8 bits and which identifies the circuit to which the mini cell belongs. Also in the header there is a length extension qualifier field 25, LEQ field, and a header integrity field 26, HIC field, both 2 bits long.

In accordance with the invention the length extension qualifier LEQ 25 is defined as a length extension for the payload and as a header extension. When LEQ takes the binary codes of 00, 01 and 10 the mini cell has the basic format shown in Figure 9 and the code bits of LEQ constitute bits to be appended to the LEN field 23. In this case the LEQ field will thus serve as an extension of the LEN field 23.

In particular, 2^4 different values in LEN field 23 is associated to the binary 00 code existing in the LEQ field 25, 2^4 different values in LEN field 23 is associated with the binary 01 code existing in LEQ field 25 and 2^4 different values in LEN 23 is associated with the binary code 10 existing in LEQ field 25. This is illustrated in Figure 10. This gives a total of 48 different length values in accordance with the following general expression:

$$[2^{\text{length of LEQ in bits}}-m] \times [2^{\text{length of LEN in bits}}],$$

where m is the number of codes used to indicate the extended format of the mini cell.

Accordingly the payload size can be chosen from forty-eight length values. In the example given the length values are coded as 1 to 48.

When the LEQ field 25 takes the binary code 11 this signifies that the basic cell format should be extended. The extended format is shown in Figure 11. The LEQ field 25 has a double meaning. The double meaning of LEQ is (i) it is used as the two most significant bits of length indication, i.e. LEQ x